

# Comparison of State-of-the-Art Deep Learning APIs for Image Multi-Label Classification using Semantic Metrics

Adam Kubany,\* Shimon Ben Ishay, Ruben Sacha Ohayon,  
Armin Shmilovici, Lior Rokach, Tomer Doitshman  
Department of Software and Information System Engineering  
Ben-Gurion University of the Negev, Israel

## Abstract

Image understanding heavily relies on accurate multi-label classification. In recent years, deep learning algorithms have become very successful for such tasks, and various commercial and open-source APIs have been released for public use. However, these APIs are often trained on different datasets, which, besides affecting their performance, might pose a challenge to their performance evaluation. This challenge concerns the different object-class dictionaries of the APIs' training dataset and the benchmark dataset, in which the predicted labels are semantically similar to the benchmark labels but considered different simply because they have different wording in the dictionaries. To face this challenge, we propose semantic similarity metrics to obtain richer understating of the APIs predicted labels and thus their performance. In this study, we evaluate and compare the performance of 13 of the most prominent commercial and open-source APIs in a best-of-breed challenge on the Visual Genome and Open Images benchmark datasets. Our findings demonstrate that, while using traditional metrics, the Microsoft Computer Vision, Imagga, and IBM APIs performed better than others. However, applying semantic metrics also unveil the InceptionResNet-v2, Inception-v3, and ResNet50 APIs, which are trained only with the simple ImageNet dataset, as challengers for top semantic performers.

**Keywords**— image multi-label classification comparison, semantic evaluation, deep learning, image understanding

---

\*Corresponding author.

E-mail addresses: adamku@post.bgu.ac.il (A.Kubany), benishue@gmail.com (S.Ben Ishay), rubensac@post.bgu.ac.il (R.Ohayon), armin@bgu.ac.il (A.Shmilovici), liorrk@post.bgu.ac.il (L.Rokach), tomerdoi@post.bgu.ac.il (T.Doitshman)

An earlier version of this article was entitled "Semantic Comparison of State-of-the-Art Deep Learning APIs for Image Multi-Label Classification."

# 1 Introduction

Accurate semantic identification of objects, concepts, and labels from images is one of the preliminary challenges in the quest for image understanding. The race to achieve accurate label classification has been fierce and became even more so as a result of public competitions such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015), and the release of benchmark datasets such as the YFCC100M (Thomee et al., 2016), Visual Genome (Krishna et al., 2017), MS-COCO (Lin et al., 2014), and Open Images (Kuznetsova et al., 2020). Different learning approaches for multi-label classification have been suggested to answer this call. Tsoumakas and Katakis (Tsoumakas & Katakis, 2006; Tsoumakas, Katakis, & Vlahavas, 2009) divided these approaches into two main categories: 1) *problem transformation methods* which transform the problem into one or more single-label classification problem and then aggregate the results into a multi-label representation; and 2) *algorithm adaptation methods* which solve the multi-label prediction problem as a whole, directly from the data. In 2012, Madjarov et al. (Madjarov, Kocev, Gjorgjevikj, & Džeroski, 2012) introduced a third category of methods, referred to as *ensemble methods*, which combine several classifiers to solve the multi-label classification problem. In this approach, each of the base classifiers in the ensemble can belong to either the problem transformation or algorithm adaptation method category.

As the research field of multi-label classification advances, more and more effective approaches have been suggested (Madjarov et al., 2012; Nasierding & Kouzani, 2012). In recent years, deep learning methods, such as convolutional neural networks (CNN), and their variations, have demonstrated excellent performance (He, Gkioxari, Dollár, & Girshick, 2017; He, Zhang, Ren, & Sun, 2015; Huang, Wang, Wang, & Tan, 2013; Ren, He, Girshick, & Sun, 2015; Thomason, Venugopalan, Guadarrama, Saenko, & Mooney, 2014; Tran et al., 2016; Vinyals, Toshev, Bengio, & Erhan, 2015; Wang et al., 2016; Yeh, Wu, Ko, & Wang, 2017). Some of the more salient approaches were published as open-source or as commercial APIs, such as from Imagga (Imagga, 2020), IBM Watson (IBM, 2020), Clarifai (Clarifai, 2020), Microsoft Computer-Vision (Microsoft, 2020), Wolfram Alpha (Wolfram, 2020), Google Cloud Vision (Google, 2020), DeepDetect (DeepDetect, 2020), YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016), MobileNet (Howard et al., 2017), Inception (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2015), ResNet (He, Zhang, Ren, & Sun, 2016) and InceptionResNet (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017). With these recent publications, the need for a best-of-breed performance comparison has arisen. While some comparisons between multi-label classification methods have been performed in the past (Madjarov et al., 2012; Nasierding & Kouzani, 2012), none of them included both open-source and commercial APIs in such extensive manner.

In this study, we address this need and evaluate the performance of 13 state-of-the-art deep learning approaches with well-established multi-label evaluation metrics (Sorower, 2010; Tsoumakas et al., 2009). While these metrics evaluate the performance based on whether the predicted labels exist in the ground truth list, they do not consider the semantic similarity between them. With this oversight, a fair comparison between the various APIs becomes challenging, as each of them can be trained on different datasets, and therefore include different object class dictionaries. To comply with this challenge, we propose to use semantic variations

of traditional evaluation metrics, the word mover’s distance metric (WMD) (Kusner, Sun, Kolkin, & Weinberger, 2015), and state-of-the-art text embedding methods such as BERT (Devlin, Chang, Lee, & Toutanova, 2018), RoBERTa (Liu et al., 2019), and XLNet (Z. Yang et al., 2019) as more insightful evaluation metrics. To the best of our knowledge, this study provides the most thorough evaluation of state-of-the-art deep learning multi-label image classification from both commercial and open-source APIs, and the only study to include semantic evaluation metrics.

The novel contributions of this study<sup>1</sup> are 1) demonstrating the significance of the proposed semantic similarity metrics to the APIs’ performance evaluation in particular when trained with different object class dictionaries, and 2) an extensive comparison of the predictive performance of 13 of the most prominent commercial and open-source publicly available APIs for multi-label image classification.

## 2 Multi-Label Image Classification APIs

We divide the classification APIs into two categories, commercial and open-source (Table 1). While the open-source APIs publish their network architecture, training schemes, and even make pre-trained models available for free use, the constantly improving commercial APIs do not reveal much about their proprietary algorithm, other than mentioning that they are based on deep neural networks.

**Commercial services:** These APIs are provided by various companies such as Imagga (Imagga, 2020), IBM Watson (Visual Recognition) (IBM, 2020), Clarifai (Clarifai, 2020), Microsoft (Computer Vision) (Microsoft, 2020), Wolfram Alpha (Image Identification) (Wolfram, 2020), and Google (Cloud Vision) (Google, 2020). Among those, only the Microsoft’s Computer Vision API hint that it is based on a deep residual network (ResNet) (He et al., 2016), which has shown high performance in the past, nevertheless, they do not reveal the network size, applied training data or any other specific variations.

We also include several top performing open-source frameworks with the capability of multi-label classification.

**DeepDetect:** The DeepDetect approach (DeepDetect, 2020) is based on the GoogLeNet architecture with 22 layers network (Szegedy, Liu, et al., 2015) also known as the Inception-v1 network. Here, we evaluate the model provided by the Caffe framework which is pre-trained on the ImageNet dataset.

**VGG19** The very deep CNN also know as the ”VGG” network (Simonyan & Zisserman, 2014), is consisted of 16-19 CNN layers. Here we included the 19 layer version trained on the ImageNet dataset (*Keras Applications*, 2020).

**Inception v3:** The Inception-v3 approach (Szegedy, Vanhoucke, et al., 2015) implements variations of the inception-v1 blocks for accuracy optimization. We evaluated the Inception-v3 Keras implementation trained on the ImageNet dataset (*Keras Applications*, 2020).

**InceptionResNet v2:** The InceptionResNet-v2 approach includes the Inception-v4 advances together with residual connections (Szegedy et al., 2017). Here, we used the ImageNet pre-trained model available via Keras implementation (*Keras Applications*, 2020).

**ResNet50:** The Residual network (ResNet) approach implements residual connec-

---

<sup>1</sup>The the APIs inference scripts and metrics applied in this study are available in [https://github.com/Adamkubany/Multilabel\\_Semantic\\_API\\_comparison](https://github.com/Adamkubany/Multilabel_Semantic_API_comparison)

tions along traditional CNN. The ResNet offers various layer depths (50, 101, 152), here, we evaluated the very popular 50 layers network trained on the ImageNet dataset (*Keras Applications*, 2020) and on the COCO dataset (Lin et al., 2014; Olafenwa & Olafenwa, 2020) as a performance reference.

**MobileNet v2:** This is the second version (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) of the MobileNet approach for mobile devices (Howard et al., 2017), it includes compact convolutional building blocks accompanied with residual ideas. We evaluated its ImageNet pre-trained model (*Keras Applications*, 2020).

**YOLO v3:** You Only Look Once (YOLO) third version approach (Redmon et al., 2016; Redmon & Farhadi, 2018) includes 106 convolution layers with residual connections and classify the image object in three scales. Here, we evaluated the DarkNet53 version pre-trained on the ImageNet dataset (Redmon & Farhadi, 2020) and on the COCO dataset (Olafenwa & Olafenwa, 2020) for additional performance reference.

### 3 Evaluation Metrics

Evaluating the different APIs’ prediction performance requires standardized measures and metrics. Various metrics have been proposed in the past for such evaluation (Sorower, 2010; Tsoumakas et al., 2009); these metrics can be divided into *bipartition* and *ranking* metrics (Tsoumakas et al., 2009). As none of the evaluated APIs provide a ranking for all of the labels in the ground truth dataset, we focus only on the bipartition metrics. For the metrics’ definitions let us denote  $Y_i \in L = \{0, 1\}^q$  as the multi-label binary encoding label set of image  $i$  from  $n$  images dataset where  $L$  is the  $q$  sized label set dictionary, and  $Z_i$  as the multi-label binary encoding label set of image  $i$  as predicted by the multi-label classifier  $h$ ; hence,  $Z_i = h(x_i) \in L$ , where  $x_i \in \mathcal{X}$ , is defined as the feature vector of image  $i$ .

#### 3.1 Bipartition Metrics

There are two types of bipartition evaluation metrics. Example-based bipartition evaluation metrics refer to various average differences of the predicted label set from the ground truth label set for all the dataset samples, whereas the label-based evaluation metrics evaluate each label separately and then average for all the labels.

##### 3.1.1 Example-Based

The following *Accuracy*, *Precision*, *Recall*, and  $F_1$  metrics are standard metrics adapted for multi-label classification (Godbole & Sarawagi, 2004; Madjarov et al., 2012). Accuracy is defined as the Jaccard similarity between the predicted label set  $Z_i$  and the ground truth label set  $Y_i$ , which is then averaged over all  $n$  images.

$$Accuracy = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (1)$$

*Precision* and *Recall* are defined as the average proportion between the number of correctly predicted labels ( $|Y_i \cap Z_i|$ ) and either the number of predicted labels  $Z_i$

or the number of ground truth labels  $Y_i$ .

$$Precision = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (2)$$

$$Recall = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i \cap Z_i|}{|Y_i|} \quad (3)$$

and  $F_1$  is the harmonic mean between *Precision* and *Recall*.

$$F_1 = \frac{1}{n} \sum_{i=1}^n \frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \quad (4)$$

### 3.1.2 Label-Based

Label-based metrics evaluate the performance of a classifier by first evaluating each label and then obtaining an average of all of the labels. Such averaging can be achieved by one of two conventional averaging operations, namely *macro* and *micro* averaging (Y. Yang, 1999). For that purpose, any binary evaluation metric can be applied, but usually *Precision*, *Recall*, and their harmonic mean  $F_1$  are applied in information retrieval tasks (Tsoumakas et al., 2009).

For each label  $\lambda_j : j = 1 \dots q$ , the summation of true positives ( $tp_j$ ), true negatives ( $tn_j$ ), false positives ( $fp_j$ ), and false negatives ( $fn_j$ ) are calculated according to the classifier applied. Then, the binary performance evaluation metric  $B$  can be calculated with either macro or micro-averaging operations:

$$Macro\ B = \frac{1}{q} \sum_{j=1}^q B(tp_j, tn_j, fp_j, fn_j) \quad (5)$$

$$Micro\ B = B \left( \sum_{j=1}^q tp_j, \sum_{j=1}^q tn_j, \sum_{j=1}^q fp_j, \sum_{j=1}^q fn_j \right) \quad (6)$$

Therefore, the definitions of *Precision* ( $P$ ), *Recall* ( $R$ ), and  $F_1$  are easily derived as (Madjarov et al., 2012):

$$Micro\ Precision\ (MiP) = \frac{\sum_{j=1}^q tp_j}{\sum_{j=1}^q tp_j + \sum_{j=1}^q fp_j} \quad (7)$$

$$Micro\ Recall\ (MiR) = \frac{\sum_{j=1}^q tp_j}{\sum_{j=1}^q tp_j + \sum_{j=1}^q fn_j} \quad (8)$$

$$Micro\ F_1 = \frac{2 \times MiR \times MiP}{MiR + MiP} \quad (9)$$

$$Macro\ Precision = \frac{1}{q} \sum_{j=1}^q \frac{tp_j}{tp_j + fp_j} \quad (10)$$

$$Macro\ Recall = \frac{1}{q} \sum_{j=1}^q \frac{tp_j}{tp_j + fn_j} \quad (11)$$

$$Macro F_1 = \frac{1}{q} \sum_{j=1}^q \frac{2 \times R_j \times P_j}{R_j + P_j}; R_j = \frac{tp_j}{tp_j + fn_j}, P_j = \frac{tp_j}{tp_j + fp_j} \quad (12)$$

where  $Macro F_1$  is the harmonic mean of *Precision* and *Recall* based on first averaging each label  $\lambda_j$  and then averaging over all labels. On the other hand,  $Micro F_1$  is the harmonic mean of *Micro Precision* and *Micro Recall* as defined above.

For all of the above metrics, they score on a scale of zero to one, where a higher score implies better alignment between the predicted label set and the ground truth set.

### 3.2 Semantic Similarity

The current formulations of the above metrics share a significant drawback as they consistently overlook the inherent semantic similarity between each label. For example, let’s assume the ground truth multi-label set is {"bicycle," "child," "helmet," "road," "tree"}, and the predicted set is {"bike," "boy," "trail," "tree," "grass," "flower"}. Evaluating the similarity between the two label sets with the above metrics will consider only the label "tree" as a true positive and overlook the close semantic similarity between the labels {"child," "boy"}, {"bicycle," "bike"} and {"road," "trail"}.

To overcome this misrepresentation, a straightforward adjustment can be made. For each of the above example-based metrics (*Accuracy*, *Precision*, *Recall*, and  $F_1$ ), the correct predictions can be decided not by the exact predicted label, but rather the semantic similarity between the predicted and true labels. Here, we use the cosine similarity ( $p \cdot r / \|p\| \|r\|$ ) between the word2vec embeddings (Mikolov, Chen, Corrado, & Dean, 2013) of the predicted ( $p$ ) and real ( $r$ ) labels, where the correct prediction is considered above certain threshold.<sup>2</sup>

Additionally, we applied the word mover’s distance (WMD) metric (Kusner et al., 2015), which is an earth mover’s distance based method (Pele & Werman, 2008, 2009), and aimed at evaluating the semantic distance between two documents. Let us denote  $Y_i^* = y_{i,j} : j = 1, \dots, r$  as the ground truth label set of image  $i$ , and  $Z_i^* = z_{i,s} : s = 1, \dots, p$  as the label set of image  $i$  predicted by the multi-label classifier  $h$ ,  $Z_i^* = h(x_i)$ . Note that  $Y_i^*$  and  $Z_i^*$  include the explicit label set (e.g., {"bike," "boy," "trail," "tree," "grass," "flower"}), where  $r$  and  $p$  don’t have to be on the same size. Defining the two label sets as two bag-of-words (BOW) allows us to apply the WDM method to evaluate their semantic distance. The WDM algorithm requires that the two BOW are represented as a normalized BOW (nBOW) vector  $d \in \mathbb{R}^n$ , where  $n = r \cup p$ , and  $d_l = t_l / \sum_{k=1}^n t_k : t_l$  is the number of times that the word  $l$  of  $n$  appears in the BOW. Let  $d$  be the nBOW representation of  $Y_i^*$  and  $d'$  of  $Z_i^*$ . The second requirement of the WDM is a semantic distance evaluation between every two labels, where  $c(l,k)$  is referred to as the cost of "traveling" from word  $l$  to word  $k$ . Therefore, Let  $W \in \mathbb{R}^{dim \times n}$  be the word2vec embedding matrix, where  $w_k \in \mathbb{R}^{dim}$  is the dim-dimensional embedding representation of word  $k$  from the vocabulary of  $n$  words. Hence, the "traveling cost" from word  $l$  to word  $k$  is defined as their Euclidean distance,  $c(l,k) = \|w_l - w_k\|$ . Next, let us define a sparse flow matrix  $T \in \mathbb{R}^{n \times n}$ , where  $t_{l,k} \geq 0$  represents the ratio of participation of word  $l$  from  $d$  to travel to word  $k$  from  $d'$ . It is clear that a word can participate in traveling as much as its nBOW  $t_l$  ratio, therefore the  $\sum_k t_{l,k} = d_l$  and  $\sum_l t_{l,k} = d'_k$

<sup>2</sup>Here, the threshold is set to 0.4.

participation ratio restrictions are applied. Finally, the distance between the two BOW can be defined as the minimum sum of the weighted traveling cost from  $d$  to  $d'$

$$wdm = \min \sum_{l,k=1}^n t_{l,k} c(l, k) \quad (13)$$

subject to the participation ratio restrictions. Since the  $wmd$  calculate the minimum traveling distance a score of 0 is considered as a perfect match. For our purposes, we average the  $wmds$  for all of the  $n$  images in the tested dataset for every API:

$$WMD = \frac{1}{n} \sum_{i=1}^n wdm_i \quad (14)$$

The above metrics are based on a single word embedding to evaluate the labels' semantic similarity; while the semantic example-based metrics use the embeddings within known metrics, the WMD takes it one step further, and considers the aggregated similarity between the ground-truth and predicted BOWs. As aggregated understanding can be beneficial for semantic similarity (Kubany, Rokach, & Shmilovici, 2020), we propose to leverage the aggregated embeddings of the BOWs as a means to find their aggregated similarity.<sup>3</sup> The BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and XLNet (Z. Yang et al., 2019) are bidirectional transformer-based methods and considered as the state-of-the-art approaches to embed an entire text to a single embedding (Wolf et al., 2019). We also consider the semantic similarity fine-tuned versions of BERT and RoBERTa<sup>4</sup> methods as they demonstrate superior performance in semantic similarity tasks (Reimers & Gurevych, 2019).

## 4 Results and Discussion

### 4.1 Experiment Setup

**Testing Dataset:** The testing dataset should include images from multiple domains, as well as multiple semantic annotations of objects, concepts, or labels, to ensure as close to real-life evaluation as possible. Some of the commercial APIs apply limits regarding the number of image requests for multi-label classification during a period of time and in total. Given these limitations, we evaluated the APIs' performance with the first 1,000 images<sup>5</sup> from each dataset, which, to our understanding, are sufficient for satisfactory performance evaluation of the examined APIs.

- **Visual Genome dataset:** The Visual Genome (VG) dataset<sup>6</sup> (Krishna et al., 2017) consists of 108,077 everyday multi-domain images, which represent the intersection between the MS-COCO (Lin et al., 2014) and the YFCC100M (Thomee et al., 2016) datasets. Each image in the dataset is associated with an average of 21 objects (out of 75,729 possibilities) for multi-label classification

<sup>3</sup>Calculated by the cosine similarity.

<sup>4</sup>Here we used the 'bert-base-nli-stsb-wkpooling' and 'roberta-base-nli-stsb-mean-tokens' pretrained versions.

<sup>5</sup>Sorted in name ascending order. We selected the first 1,000 images that have objects, as some of the images do not have them.

<sup>6</sup>We used the 1.0 version of the dataset.

purposes. Within the 1000 images subset, there are 3728 possible objects, an average of 14.1 objects per image, and 1.05 labels per object, where 8.5% of the used labels are unknown to the word2vec embedding model.

- **Open Images dataset:** The Open Images (OI) dataset<sup>7</sup> (Krasin et al., 2017; Kuznetsova et al., 2020) incorporate  $\sim 9$ M images of diverse sceneries collected from the "Flickr" online service. For multi-label classification, each image includes an average of 8.3 objects out of 600 classes. Within the 1000 subset, each image includes an average of 3.9 objects out of 263 object classes, where 16.5% of the 3.9 average objects are unknown to the word2vec embedding model.

It is essential to consider the applied training data for the various APIs. We assume that the commercial APIs vendors continuously attempt to improve their services. Since the commercial APIs' training data is unknown, selecting widely accessible and popular datasets makes it more likely to be considered, and therefore should confine the predicament of biased evaluation. For the open-source APIs, we chose the ImageNet pre-trained models, which is well-known simple scenery dataset and should provide as an adequate baseline.

**APIs' Evaluated Objects:** Some of the commercial APIs restrict the number of predicted objects per image, while others predict only a few object labels with high confidence and low confidence for others. Evaluating with different top levels allow a fair comparison between the APIs. In this study, we perform the APIs evaluation based on three object levels: the top five, three, and one label(s) according to their confidence level (see Tables 3-9). Also, for a fair comparison, we queried all the APIs using their vanilla versions without any specific fine-tuning.

API	Type	Training Data	Unknown Labels out of Top Five Objects (%) (VG / OI)	Mean Labels Per Object (VG / OI)
Clarifai	Commercial	unknown	0.5 / 6.5	1 / 1
Google Cloud Vision	Commercial	unknown	10 / 15.4	1 / 1
IBM Watson	Commercial	unknown	0.8 / 8.1	1 / 1
Immaga	Commercial	unknown	5.6 / 5.7	1 / 1
Microsoft Computer Vision	Commercial	unknown	1.1 / 2.7	1 / 1
Wolfram	Commercial	unknown	42.3 / 23.8	1 / 1
DeepDetect	Open Source	ImageNet	10.4 / 11	1.95 / 1.89
InceptionReNet-v2	Open Source	ImageNet	11.9 / 11.1	1.98 / 1.95
Inception-v3	Open Source	ImageNet	12.1 / 11.4	1.97 / 1.94
MobileNet-v2	Open Source	ImageNet	11.4 / 10.6	1.97 / 1.93
ResNet50	Open Source	ImageNet	10.3 / 9.2	1.99 / 1.95
ResNet50	Open Source	COCO	5.2 / 3.9	1 / 1
VGG19	Open Source	ImageNet	9.6 / 10.2	2 / 1.93
YOLO-v3	Open Source	ImageNet	19.6 / 16	1 / 1
YOLO-v3	Open Source	COCO	5.5 / 4	1 / 1

Table 1: APIs' metadata.

<sup>7</sup>We used the sixth version of the dataset.

## 4.2 Example-Based Metrics

One of the first observations is that in general, the examined APIs have relatively low scores. A few factors can explain this observation; first, there is the issue of model settings and training data, and although we apply well-known datasets to reduce testing bias, we do not know which training data was used by the commercial APIs, on the other hand, the open-source APIs behave as expected as they all were pre-trained with the simple ImageNet dataset. Nevertheless, since the commercial APIs achieve higher scores than the open-source APIs, it might suggest that at least some images of the datasets were in their training data. Additionally, although the commercial APIs' out of the box configurations also contribute to the low scores, they are necessary if we wish for a fair comparison without prior knowledge of its structure. Hence, we analyze the commercial and open-source APIs separately.

Second, the VG dataset holds an average of 14.1 objects per image, while we account for a maximum of five predictions, this explains the low scores of the *Accuracy*, *Recall*, and  $F_1$  metrics, as they consider the number of the ground truth objects' labels. On the other hand, the OI dataset holds an average of 3.9 objects per image, thus, provide a larger scale of *Accuracy*, *Recall*, and  $F_1$  metrics' scores, with the same scale for the *Precision* metric.

**Commercial APIs:** The commercial APIs' performance is consistent on both datasets, and reveal that four APIs stand out with high scores: Microsoft Computer Vision (MCV), Imagga, IBM and Google APIs consistently hold top places, with the MCV API dramatically outperform others (inhabit the most of green cells in Tables 3-9). It is worth noting that the Google API performance is not consistent between the datasets, as it holds a higher place in the OI dataset. We might explain this top performance to the fact that Google also manufactures the OI dataset, and it could have been used in its training. Having a high *Precision* score means that most of the predictions made by the MCV API are relevant, with only a few false positives. It also has a high *Recall* scores, indicating it predicted relatively more of the ground truth labels (with only a few false negatives); this is also reflected in the relatively high *Accuracy* score. The MCV's top  $F_1$  score also reassures its dominance as it is the harmonic mean of its high scores in the *Precision* and *Recall* metrics. Since the number of true labels is higher than the predicted ones, in our view, the *Precision* metric gives a more reliable indication of the APIs performance. Here, the MCV's *Precision* dramatically outperforms others, which means that its order of predictions is closer to the true labels than other APIs.

Another perspective is the score dynamic between the different top predicted levels. As expected, as the number of predicted labels rise, the *Precision* scores decrease (Figures 5 and 8) as it is more likely to be correct in one label than in five, and the *Recall* scores increase (Figures 4 and 7) as it more likely to find more labels in common with the true labels. The increasing  $F_1$  scores (Figures 3 and 6) teach us that the *Recall* dynamic change is stronger than the *Precision* one.

**Open-Source APIs:** As expected, the open-source APIs produce much lower results than the commercial APIs. These relatively low results can be explained by their training data, as they are all trained on the ImageNet dataset, which includes much simpler sceneries and different labels than the datasets. With that saying, the APIs consisted of more elaborated network architecture yield better performance, usually in relation to their specific engineering advances, with the general performance order of InceptionResNet-v2, Inception-v3, MobileNet-v2, ResNet50, YOLO-v3, and VGG19. As before, we particularly notice the performance

differences within the *Precision* metric, which demonstrate the InceptionResNet-v2 and Inception-v3 dominance. We will revisit the analysis of these APIs within the semantic analysis (section 4.4).

The superior performance of the commercial APIs, and in particular, of the MCV API raises the question of whether their top performance is due to their network structure, training data, or both. We can only partially answer this question since the complete details of commercial APIs is unknown; fortunately, we know that the top-performing MCV API is based on the ResNet architecture (He et al., 2016), but unsure of its specific network details and training data. Therefore, to confine this question, we compare its performance on the VG dataset with the ResNet50 and YOLO-v3 APIs trained on both ImageNet and COCO<sup>8</sup> datasets (Table 6). We can see that the performance change of the ResNet50 and YOLO-v3 APIs between the ImageNet and the COCO pre-trained models are consistent in scale. Let us analyze the MCV API performance under the assumption that Microsoft would offer the best possible network architecture they have in their arsenal for their payable API. Considering that, if the MCV would have been trained only on the COCO dataset, it needed to outperform the COCO pre-trained ResNet50, as it is the leanest flavor of ResNet with 50 layers, where deeper and better-performing networks with 101 and 152 layers exist (He et al., 2016). Since, in general, the MCV yield scores higher than than the ImageNet pre-trained ResNet50 and lower than the COCO pre-trained ResNet50, in particular within the top one and three predictions, we are left to conclude that it is not trained solely on the COCO dataset, and can remain in the evaluation. Also, we perform the same analysis of the MCV, ResNet50, and YOLO-v3 performance on the OI dataset (Table 10). Since, as far as we know, the OI does not include the COCO dataset, and the MCV API dominant over the ResNet50 COCO API, it reassures our previous conclusion that the MCV API is trained on more images than those inhabit the COCO dataset.

There is an immense performance jump when training with the COCO dataset for the ResNet50 and YOLO-v3 APIs on both datasets. The performance turnover is so significant for the YOLO-v3 API on the OI dataset, that it changes its rank from the last place to among top performers, especially in the top predicted label. This performance change highlights the drawback of training with a simple dataset as the ImageNet and makes us wonder about the potential performance improvement of the other open-source APIs, which demonstrate better performance than YOLO-v3 when trained only on the ImageNet dataset.

From the example-based metrics perspective, the MCV is the top all-around performer; nevertheless, if other APIs are needed, the Imagga and IBM are excellent options.

### 4.3 Label-Based Metrics

Within this type of metrics, we evaluate the performance of the various APIs from the label perspective (see Tables 3-9). In the *Macro* family of metrics, we evaluate the performance of predicting each label separately and then averaging them all, whereas, in the *Micro* metrics, we evaluate the performance of all the labels' predictions together. Furthermore, since the *Macro* metrics does not account for the false predictions (*fp* for *Precision* and *fn* for *Recall*) when the true positive

---

<sup>8</sup>The VG dataset in the intersection between the MS-COCO and the YFCC100M datasets.

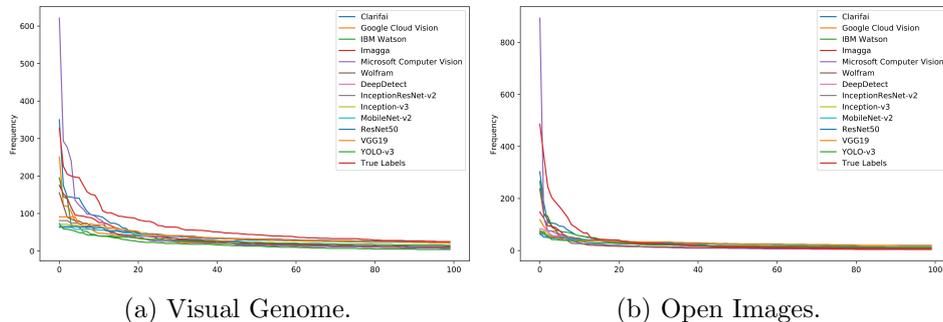


Figure 1: 100 most Frequent object classes.

is zero, and the *Micro* metrics does, we can evaluate the prediction balance between the labels. We notice low *Macro* scores for the VG dataset, whereas the OI dataset allows much higher scores. The VG low scores suggest that many labels have zero true positives, which agree with a higher number of object classes (3728 in the VG, and 263 in the OI) and a long tail object class distribution (Figure 1). The score difference of the *Recall* metric in between the VG and OI datasets, continue to support this line of thinking. The same score scale of the *Recall Micro* and *Recall Macro* metrics, with even a slightly higher *Recall Micro* score in the VG dataset, further indicates that the zero true positives are of infrequent labels. On the other hand, while the score scale of the *Precision* metrics in the OI dataset is about the same, in the VG dataset, it is not. As before, The VG’s low *Precision Macro* score is due to the many zero true positives, while the higher *Precision Micro* score indicates that there are less false positives. Still, the *Precision Micro* score in the VG dataset is lower than in the OI dataset due to the division in dataset’s object class number. **Commercial APIs:** Like with the *example-based* metrics, the MCV, IBM, and Imagga APIs, stand out on both datasets. However, the Google Cloud Vision APIs, which demonstrate only occasional good performance on the VG dataset, outperform all APIs in the OI dataset. As before, we suspect that the exceptional top scores of Google Cloud Vision API on the OI dataset might suggest that the Google published dataset is part of its training set.

**Open-Source APIs:** For these APIs, besides within the *Micro Precision* metric, the performing differences between the best and worst APIs are marginal, making it very hard to gain any knowledge. Nevertheless, there are two points worthy of pointing out; first, within the *Macro Recall* metric, for the first time, although in a small margin, the InceptionResNet-v2 and the Inception-v3 APIs consistently outperform all others in the VG dataset, meaning that on average they are slightly more capable of predicting the correct label. Second, for the *Micro Precision* metric, the YOLO-v3 API performs better than other open-source APIs on the VG dataset, following by the InceptionResNet-v2, Inception-v3, and ResNet50 APIs.

The MCV, Imagga, IBM, and Google APIs are ahead with the MCV outperforming all other considering the scores from both the example and label-based metrics. Additionally, the InceptionResNet-v2 and the Inception-v3 APIs demonstrate some good performance, and it would be wise to give them further consideration.



Figure 2: Example image '1.jpg' from the Visual Genome dataset

API	Labels	Recall	Recall (Semantic)	Precision	Precision (Semantic)	WMD	Fine-Tuned BERT	Fine-Tuned RoBERTa
Clarifai	city, <u>vehicle</u> , people, road, <b>street</b>	0.04	0.08	0.2	0.4	3.19	0.28	0.13
Google Cloud Vision	city, pedestrian, <b>street</b> , signage, walking	0.04	0.08	0.2	0.4	3.48	0.27	0.19
IBM Watson	<b>street</b> , city, crowd, people	0.04	0.04	0.25	0.25	3.35	0.13	-0.02
Imagga	<b>sidewalk</b> , walk, <b>street</b> , road, city	0.08	0.08	0.4	0.4	3.30	0.29	0.20
Microsoft Computer Vision	outdoor, <b>building</b> , <b>street</b> , road, <b>sidewalk</b>	0.12	0.12	0.6	0.6	3.20	0.28	0.29
Wolfram	road, path, container, conveyance, <u>vehicle</u>	0	0.04	0	0.2	3.62	0.25	0.02
DeepDetect	ski, crutch, [prison, prison house], sliding door, shovel	0	0	0	0	3.69	0.25	0.21
InceptionResNet-v2	<b>parking meter</b> , [traffic light, traffic signal, <u>stoplight</u> ], [pay-phone, pay-station], [mailbox, letter box], [cash machine, cash dispenser, automated teller machine]	0.04	0.08	0.2	0.4	3.80	0.35	0.25
Inception-v3	[jirikisha, ricksha, <u>rickshaw</u> ], [ashcan, trash can, garbage can, wastebin], [streetcar, tram, trolley, trolley car], [bookshop, bookstore, bookstall], plastic bag	0	0.04	0	0.2	3.77	0.48	0.30
MobileNet-v2	<b>parking meter</b> , [jirikisha, ricksha, <u>rickshaw</u> ], [police van, police wagon, paddy wagon], [cab, hack, taxi, taxicab], crutch	0.04	0.12	0.2	0.6	3.57	0.45	0.26
Resnet50	<b>parking meter</b> , [mailbox, letter box], [ashcan, trash can, garbage can, wastebin], [traffic light, traffic signal, <u>stoplight</u> ], [jirikisha, ricksha, <u>rickshaw</u> ]	0.04	0.12	0.2	0.6	3.69	0.47	0.29
Resnet50 (COCO)	<u>person</u> , <u>car</u> , bicycle, traffic light, truck	0.04	0.12	0.2	0.6	3.20	0.35	0.24
VGG19	<b>parking meter</b> , [jirikisha, ricksha, <u>rickshaw</u> ], [gas pump, gasoline pump, petrol pump], ski, ambulance	0.04	0.08	0.2	0.4	3.78	0.34	0.25
YOLO-v3	pole, cash machine, guillotine, plastic bag, <u>jean</u>	0	0.02	0	0.2	3.83	0.26	0.07
YOLO-v3 (COCO)	<u>person</u> , <u>car</u> , truck, bicycle, <b>parking meter</b>	0.08	0.16	0.4	0.8	3.26	0.36	0.17
Ground Truth	arm, back, bike, bikes, building, car, chin, clock, glasses, guy, headlight, jacket, lamp post, man, pants, parking meter, shade, shirt, shoes, sidewalk, sign, sneakers, street, tree, tree trunk							

Table 2: The APIs' top five labels for image '1.jpg' from the Visual Genome dataset (Figure 2). The table's annotations refer to **bold** labels as *tp* and the underline labels as semantic similar *tp*.

## 4.4 Semantic Metrics

To demonstrate the importance of semantic metrics within the multi-label evaluation, we refer to the APIs' top five predicted labels of image 1 from the VG dataset as an example (Figure 2, Table 2). Within this image, let us take the first API as an example, in human prospective the Clarifai API includes many valid labels, but only the "street" label is correct according to the ground-truth labels list. For instance, the "vehicle" label is not included in the ground-truth label set, and traditionally not considered as a true positive prediction. Obviously, it has the same meaning as the ground-truth "car" label, and a similarity score of 0.78, and therefore, is considered as semantic true positive, as it should be. Since we are more interested in the semantic meaning of the labels rather than their exact wording, this example, like many others, showcase the necessity of semantic metrics for multi-label evaluation, in particular when the APIs are trained on a different object class list from the examined dataset.

The word2vec embedding model is well-known to be an accurate and comprehensive word representation. Nevertheless, despite our efforts,<sup>9</sup> some of the APIs' predicted labels are not found within the model and considered as the "unknown" label (see Table 1 for APIs' "unknown" rates). We further discuss the effect of these settings with regard to semantic example-based and WMD metrics.

**Semantic Example-Based Metrics:** Evaluating the APIs' performance with the classic approach stated the MCV, IBM, and Imagga APIs as top performers with the MCV outperforming all, the semantic evaluation on the VG dataset shed a new light of the APIs performance. In this evaluation, the MCV is still a high performer, but now the InceptionResNet-v2, Inception-v3, and frequently the ResNet50 and MobileNet-v2 APIs demonstrate top performance, meaning they can predict more closely semantically related labels. These findings are even more dramatic as they are trained on the ImageNet dataset and have about ten percent of "unknown" labels (Table 1), even if considering that they predict about twice the labels per object, their rate of unknowns is still higher. A higher percentage of "unknowns" makes it harder to recognize a predicted label as true positive, which lowers the performance metrics score. These findings demonstrate the performance advantage of a more elaborated network structure over simpler and shallower networks (like the DeepDetect API); moreover, it highlights the importance of the residual ideas as it exists in three of the five top semantic performers (InceptionResNet-v2, MCV, and ResNet50). The semantic evaluation on the OI dataset incline to less dramatic results for the open-source APIs in terms of top performers. Nevertheless, the simple dataset trained APIs, which naturally achieve much lower scores than the commercial APIs with the traditional metrics, now measure on the same semantic score scale. We relate the lower open-source semantic scores in the OI dataset to its lower average objects per image (14.1 in the VG vs. 3.9 in the OI), as the semantic metrics benefit from more labels to be potentially semantically similar. These results further demonstrate the benefits of the semantic metrics, which can compare APIs with a lesser effect of their training dataset.

---

<sup>9</sup>For example, the label "parking meter" for image 1, which is first lowercased and cleaned from unwanted chars, does not exist in the word2vec model. We try different permutation of the label: without space ("parkingmeter"), with an underscore ("parking\_meter"), with first caps and underscore ("Parking.Meter"). In this case, the "Parking.Meter" label permutation exists in the word2vec model.

It is important to note that these findings of the performance dominance of the more elaborated APIs (InceptionResNet-v2, Inception-v3, and frequently the ResNet50) found here are consistent with prior ImageNet evaluations (Silberman & Guadarrama, 2016), and further validate our findings.

Additionally, the high "unknown" rates of the Wolfram (42.3% / 23.8%), and YOLO-v3 (19.6% / 16%) APIs can partially explain their semantic low scores. When such a significant portion of the labels predicted as "unknown" and different from the actual labels, it makes no surprise they have such low scores.

**WMD Metric:** Following the superior performance of the InceptionResNet-v2, Inception-v3, ResNet50 APIs in the semantic example-based metrics, we are somewhat surprised as to their lower scores with the WMD metric. Nevertheless, in our view, a simple explanation exists. If we take the top five level case as an example, each of the open-source APIs predicts about ten labels per image (two labels per object, see Table 1), except for the YOLO-v3 API which predicts five labels. Considering their "unknown" rates, on average, all of the open-source APIs include one "unknown" label, which is much rarer in the commercial APIs. As the WMD metric calculates the minimum traveling distance between the true and predicted nBOW label vectors, the inclusion of the "unknown" label in the nBOW is forcing the semantic distance to be much higher, hence lowering the open-source APIs scores.

**Labels' BOW Embedding:** Considering a BOW aggregated embedding allows us to overcome the issue of "unknown" labels and to estimate the aggregated labels' semantic similarity directly. Within this evaluation, the vanilla version of the BERT, RoBERTa, and XLNet methods produced poor results with  $\sim 0.007$  standard deviation between the APIs scores. These results are consistent with previous findings in which their vanilla versions are prone to poor sentence embeddings (Reimers & Gurevych, 2019). In contrast, the semantic similarity scores of the fine-tuned versions of the BERT and RoBERTa methods<sup>10</sup> demonstrate conclusive results on the VG dataset. These findings demonstrate that the results of both embeddings agree with each other, and strongly supports the previous semantic example-based metrics results, in which the InceptionResNet-v2, Inception-v3, and ResNet50 APIs consistently demonstrate top performance. As with the *Semantic Example-Based* metrics, the open-source APIs perform better with VG dataset than with the OI dataset. The BERT and RoBERTa methods benefit from more input words to produce more accurate embeddings (up to a point) and the lesser amount of the OI objects per image, in particular in the face of a large amount of BOW predicted labels of the open-source APIs harm their semantic similarity score. Therefore, the performance evaluation of the open-source APIs with the OI dataset is less informative than with the VG dataset. Within this context, it is essential to remember that the issue of input word number is critical for accurate embeddings for every use in such methods.

Whereas the non-semantic metrics exhibit the dominance of the MCV, Imagga, and IBM APIs, The semantic metrics challenge their dominance and allow the simpler dataset trained APIs to be considered as equals, and even weigh the InceptionResNet-v2, Inception-v3, and ResNet50 APIs as the top semantic performers.

---

<sup>10</sup>We have not found the semantic similarity fine-tuned version of XLNet.

## 5 Conclusions

In this study, we compared the performance of some of the most prominent deep learning multi-label classification APIs. Throughout our evaluations using the traditional metrics approaches, the MCV, IBM, and Imagga APIs consistently demonstrate top performance with the MCV API as the top performer; obviously, their performance is no match for the open-source APIs which are trained with a much simpler dataset. However, the semantic metrics allow these low starting point APIs to be evaluated as equals and ever consider the InceptionResNet-v2, Inception-v3, and ResNet50 APIs among the top semantic performers. These evaluations demonstrate the capabilities and added value of the semantic metrics in obtaining profound insights regarding the labels meaning even when training with a simple dataset and a different object-class dictionary, insights, which are unavailable otherwise.

As the field of multi-label classification advances, we believe that the proposed semantic metrics and the performance comparison performed in this study can be beneficial for both researchers and users in the quest for image understanding.

## Acknowledgments

This study was supported by grants from the MAGNET program of the Israeli Innovation Authority and the MAFAT program of the Israeli Ministry of Defense.

## References

- Clarifai. (2020). *Clarifai website*. Retrieved from <https://www.clarifai.com/>
- DeepDetect. (2020). *DeepDetect*. Retrieved from <https://www.deepdetect.com/>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Godbole, S., & Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. *Advances in Knowledge Discovery and Data Mining*, 22–30.
- Google. (2020). *Google Cloud Vision*. Retrieved from <https://cloud.google.com/vision/>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034).

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Huang, Y., Wang, W., Wang, L., & Tan, T. (2013). Multi-task deep neural network for multi-label learning. In *Image processing (icip), 2013 20th IEEE international conference on* (pp. 2897–2900). IEEE.
- IBM. (2020). *Visual Recognition*. Retrieved from <https://www.ibm.com/watson/services/visual-recognition/>
- Imagga. (2020). *Imagga website*. Retrieved from <https://imagga.com/solutions/auto-tagging.html>
- Keras Applications*. (2020). Retrieved from <https://keras.io/applications/>
- Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., ... Murphy, K. (2017). OpenImages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*.
- Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., ... Fei-Fei, L. (2017, 5). Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123(1), 32–73. Retrieved from <https://visualgenome.org/> doi: 10.1007/s11263-016-0981-7
- Kubany, A., Rokach, L., & Shmilovici, A. (2020). Triplet Semantic Similarity Using Embedding’s Analogy Divergence. *Manuscript submitted for publication*.
- Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). From Word Embeddings To Document Distances. In *Icml* (Vol. 15, pp. 957–966).
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., ... Duerig, T. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In (pp. 740–755). Springer.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9), 3084–3104.
- Microsoft. (2020). *Computer-vision API website*. Retrieved from

- <https://www.microsoft.com/cognitive-services/en-us/computer-vision-api>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Nasierding, G., & Kouzani, A. Z. (2012). Comparative evaluation of multi-label classification methods. In *Fuzzy systems and knowledge discovery (fskd), 2012 9th international conference on* (pp. 679–683). IEEE.
- Olafenwa, M., & Olafenwa, J. (2020). *ImageAI - Detection*. Retrieved from <https://imageai.readthedocs.io/en/latest/detection/index.html>
- Pele, O., & Werman, M. (2008). A linear time histogram metric for improved sift matching. In *European conference on computer vision* (pp. 495–508). Springer.
- Pele, O., & Werman, M. (2009). Fast and robust earth mover’s distances. In *Computer vision, 2009 IEEE 12th international conference on* (pp. 460–467). IEEE.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (cvpr)* (pp. 779–788).
- Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Redmon, J., & Farhadi, A. (2020). *YOLOv3 - ImageNet Classification*. Retrieved from <https://pjreddie.com/darknet/imagenet/>
- Reimers, N., & Gurevych, I. (2019, 11). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing*. Association for Computational Linguistics. Retrieved from <http://arxiv.org/abs/1908.10084>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 211–252. Retrieved from <http://www.image-net.org/challenges/LSVRC/> doi: 10.1007/s11263-015-0816-y
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).
- Silberman, N., & Guadarrama, S. (2016). *TensorFlow-Slim image classification model library*. Retrieved from <https://github.com/tensorflow/models/tree/master/research/slim>

- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*.
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first aai conference on artificial intelligence* (pp. 936–940).
- Szegedy, C., Liu, W., Jia, Y., Reed, S., Anguelov, D., Erhan, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1–9).
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*.
- Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., & Mooney, R. (2014). Integrating language and vision to generate natural language descriptions of videos in the wild. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers* (pp. 1218–1227).
- Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., ... Li, L.-J. (2016). YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2), 64–73.
- Tran, K., He, X., Zhang, L., Sun, J., Carapcea, C., Thrasher, C., ... Sienkiewicz, C. (2016). Rich image captioning in the wild. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 49–56).
- Tsoumakas, G., & Katakis, I. (2006). Multi-label classification: An overview. *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*.
- Tsoumakas, G., Katakis, I., & Vlahavas, I. (2009). Mining multi-label data. *Data mining and knowledge discovery handbook*, 667–685.
- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3156–3164).
- Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification. In *Computer vision and pattern recognition (cvpr), 2016 ieee conference on* (pp. 2285–2294). IEEE.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Brew, J. (2019). HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *ArXiv, abs/1910.03771*.
- Wolfram. (2020). *Wolfram Alpha: Image Identification Project*. Retrieved from <https://www.imageidentify.com/>
- Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2), 69–90.

- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5754–5764).
- Yeh, C.-K., Wu, W.-C., Ko, W.-J., & Wang, Y.-C. F. (2017). Learning Deep Latent Space for Multi-Label Classification. In *Aaai* (pp. 2838–2844).

API	Accuracy	Accuracy (Semantic)	Recall	Recall (Semantic)	Precision	Precision (Semantic)	F1	F1 (Semantic)	Macro Precision	Macro Recall	Macro F1	Micro Precision	Micro Recall	Micro F1	WMD	Fine-Tuned BERT	Fine-Tuned RoBERTa
Clarifai	0.020	0.050	0.020	0.050	0.178	0.484	0.033	0.081	0.007	0.002	0.002	0.201	0.011	0.020	3.658	0.162	0.106
Google Cloud Vision	0.016	0.051	0.016	0.051	0.139	0.460	0.025	0.083	0.008	0.003	0.004	0.191	0.008	0.016	3.715	0.190	0.145
IBM Watson	0.026	0.055	0.026	0.055	0.214	0.442	0.042	0.088	0.008	0.004	0.004	0.293	0.018	0.034	3.728	0.178	0.156
Imgaga	0.028	0.063	0.028	0.063	0.284	0.559	0.046	0.102	0.0106	0.005	0.005	0.324	0.017	0.033	3.644	0.220	0.169
Microsoft Computer Vision	0.042	0.059	0.042	0.059	0.337	0.485	0.067	0.095	0.0108	0.002	0.003	0.324	0.020	0.038	3.614	0.195	0.111
Wolfram	0.006	0.038	0.006	0.038	0.058	0.397	0.010	0.065	0.003	0.001	0.001	0.082	0.002	0.004	3.845	0.159	0.116
DeepDetect	0.007	0.054	0.007	0.054	0.102	0.526	0.013	0.090	0.005	0.005	0.003	0.205	0.006	0.012	3.915	0.256	0.205
InceptionResNet-v2	0.013	0.065	0.013	0.065	0.158	0.594	0.023	0.106	0.008	0.008	0.006	0.262	0.009	0.018	3.553	0.283	0.225
Inception-v3	0.013	0.065	0.013	0.065	0.144	0.590	0.022	0.107	0.007	0.008	0.006	0.242	0.009	0.017	3.871	0.284	0.223
MobileNet-v2	0.010	0.062	0.010	0.062	0.131	0.587	0.018	0.102	0.007	0.007	0.005	0.241	0.008	0.016	3.869	0.274	0.223
Resnet50	0.011	0.063	0.011	0.063	0.129	0.578	0.019	0.103	0.007	0.006	0.005	0.245	0.008	0.015	3.878	0.266	0.219
VGG19	0.008	0.055	0.008	0.055	0.101	0.532	0.014	0.093	0.005	0.006	0.004	0.198	0.006	0.012	3.902	0.257	0.210
YOLO-v3	0.011	0.054	0.011	0.054	0.123	0.517	0.018	0.090	0.006	0.006	0.004	0.285	0.008	0.016	4.001	0.251	0.184

Table 3: The APIs’ metrics results with the top predicted label per image evaluated on the Visual Genome dataset. The color gradient scales the APIs performance from green (best) to red (worst).

API	Accuracy	Accuracy (Semantic)	Recall	Recall (Semantic)	Precision	Precision (Semantic)	F1	F1 (Semantic)	Macro Precision	Macro Recall	Macro F1	Micro Precision	Micro Recall	Micro F1	WMD	Fine-Tuned BERT	Fine-Tuned RoBERTa
Clarifai	0.030	0.077	0.042	0.101	0.131	0.314	0.054	0.130	0.012	0.006	0.005	0.160	0.024	0.042	3.508	0.253	0.220
Google Cloud Vision	0.031	0.092	0.043	0.119	0.133	0.396	0.055	0.156	0.012	0.008	0.007	0.196	0.024	0.044	3.531	0.311	0.268
IBM Watson	0.038	0.074	0.051	0.091	0.180	0.362	0.066	0.123	0.009	0.009	0.008	0.283	0.035	0.062	3.627	0.247	0.232
Imgaga	0.043	0.101	0.058	0.130	0.193	0.424	0.076	0.170	0.014	0.009	0.008	0.238	0.035	0.061	3.432	0.309	0.282
Microsoft Computer Vision	0.070	0.105	0.095	0.138	0.283	0.423	0.119	0.176	0.014	0.007	0.007	0.283	0.050	0.085	3.328	0.320	0.265
Wolfram	0.009	0.070	0.012	0.089	0.041	0.318	0.017	0.121	0.006	0.003	0.003	0.090	0.006	0.012	3.670	0.253	0.239
DeepDetect	0.012	0.097	0.015	0.123	0.063	0.425	0.021	0.164	0.005	0.009	0.005	0.133	0.011	0.021	3.704	0.356	0.320
InceptionResNet-v2	0.018	0.113	0.023	0.143	0.089	0.486	0.032	0.190	0.007	0.013	0.007	0.172	0.017	0.030	3.663	0.389	0.347
Inception-v3	0.016	0.109	0.021	0.139	0.081	0.471	0.029	0.183	0.007	0.011	0.007	0.158	0.015	0.027	3.664	0.385	0.340
MobileNet-v2	0.015	0.105	0.019	0.132	0.071	0.455	0.026	0.176	0.006	0.010	0.006	0.150	0.013	0.024	3.684	0.370	0.336
Resnet50	0.014	0.108	0.019	0.138	0.074	0.461	0.026	0.181	0.006	0.010	0.006	0.156	0.014	0.025	3.685	0.373	0.335
VGG19	0.012	0.100	0.015	0.128	0.062	0.432	0.021	0.169	0.005	0.009	0.005	0.128	0.011	0.020	3.706	0.360	0.323
YOLO-v3	0.013	0.086	0.017	0.109	0.070	0.390	0.024	0.148	0.005	0.008	0.005	0.186	0.014	0.025	3.798	0.356	0.321

Table 4: The APIs’ metrics results with the top three predicted labels per image evaluated on the Visual Genome dataset. The color gradient scales the APIs performance from green (best) to red (worst).

API	Accuracy		Recall		Precision		F1		Macro		Micro		WMD	Fine-Tuned		
	Top	Accuracy (Semantic)	Recall (Semantic)	Precision (Semantic)	F1 (Semantic)	Precision	Recall	F1	Precision	Recall	F1	Micro Recall		Micro F1	BERT	RoBERTa
Clarifai	0.037	0.089	0.136	0.116	0.274	0.066	0.153	0.015	0.009	0.007	0.149	0.036	0.057	3.445	0.297	0.273
Google Cloud Vision	0.034	0.105	0.155	0.114	0.348	0.062	0.179	0.016	0.013	0.010	0.185	0.033	0.056	3.480	0.351	0.318
IBM Watson	0.038	0.075	0.100	0.161	0.336	0.066	0.125	0.013	0.011	0.009	0.261	0.038	0.066	3.625	0.253	0.240
Imagga	0.049	0.113	0.167	0.158	0.352	0.088	0.192	0.013	0.013	0.010	0.204	0.048	0.077	3.365	0.343	0.330
Microsoft Computer Vision	<b>0.079</b>	0.129	<b>0.126</b>	<b>0.248</b>	0.396	<b>0.137</b>	0.214	<b>0.0181</b>	0.011	<b>0.011</b>	<b>0.266</b>	<b>0.069</b>	<b>0.110</b>	<b>3.229</b>	0.370	0.324
Wolfram	0.011	0.090	0.129	0.037	0.291	0.020	0.154	0.007	0.004	0.003	0.083	0.009	0.017	3.603	0.300	0.292
DeepDetect	0.013	0.116	0.168	0.048	0.369	0.024	0.196	0.005	0.010	0.005	0.109	0.015	0.026	3.647	0.392	0.354
InceptionResNet-v2	0.017	<b>0.130</b>	0.027	0.063	<b>0.411</b>	0.032	<b>0.217</b>	0.006	<b>0.015</b>	0.007	0.125	0.019	0.033	3.606	<b>0.423</b>	<b>0.375</b>
Inception-v3	0.017	0.126	0.027	0.184	0.400	0.031	0.212	0.006	0.014	0.006	0.127	0.018	0.032	3.616	0.418	0.371
MobileNet-v2	0.015	0.123	0.023	0.053	0.391	0.027	0.208	0.005	0.013	0.006	0.120	0.016	0.029	3.637	0.408	0.366
ResNet50	0.015	0.123	0.023	0.054	0.390	0.027	0.208	0.005	0.013	0.005	0.116	0.016	0.028	3.629	0.411	0.369
YGG19	0.013	0.119	0.020	0.175	0.376	0.024	0.200	0.005	0.011	0.005	0.107	0.014	0.025	3.646	0.398	0.354
YOLO-v3	0.015	0.104	0.024	0.055	0.337	0.028	0.178	0.005	0.012	0.005	0.149	0.017	0.031	<b>3.726</b>	0.401	0.366

Table 5: The APIs’ metrics results with the top five predicted labels per image evaluated on the Visual Genome dataset. The color gradient scales the APIs performance from green (best) to red (worst).

API	Accuracy		Recall		Precision		F1		Macro		Micro		WMD	Fine-Tuned		
	Top	Accuracy (Semantic)	Recall (Semantic)	Precision (Semantic)	F1 (Semantic)	Precision	Recall	F1	Precision	Recall	F1	Micro Recall		Micro F1	BERT	RoBERTa
Microsoft Computer Vision	1	0.042	0.059	0.042	0.495	0.067	0.095	0.011	0.002	0.0034	0.3240	0.020	0.038	3.614	0.195	0.111
ResNet50	1	0.011	0.063	0.011	0.129	0.578	0.103	0.007	0.006	0.0049	0.2445	0.008	0.015	3.878	0.266	0.219
ResNet50 (coco)	1	0.057	0.089	0.057	0.905	0.798	0.094	0.009	0.006	0.0066	0.8784	0.027	0.050	3.493	0.259	0.168
YOLO-v3	1	0.011	0.054	0.011	0.123	0.517	0.118	0.006	0.006	0.0045	0.2850	0.008	0.016	4.001	0.251	0.184
YOLO-v3 (coco)	1	0.056	0.090	0.056	0.690	0.489	0.149	0.009	0.009	0.0060	0.3397	0.024	0.045	3.459	0.241	0.147
Microsoft Computer Vision	3	0.070	0.105	0.095	0.138	0.283	0.119	0.014	0.007	0.0073	0.2830	0.050	0.085	3.328	0.320	0.265
ResNet50	3	0.014	0.108	0.019	0.138	0.074	0.181	0.006	0.010	0.0060	0.1559	0.014	0.025	3.685	0.373	0.355
ResNet50 (coco)	3	0.071	0.149	0.090	0.177	0.316	0.242	0.008	0.011	0.0089	0.3377	0.047	0.083	3.276	0.390	0.324
YOLO-v3	3	0.013	0.086	0.017	0.109	0.070	0.390	0.005	0.008	0.0049	0.1855	0.014	0.025	3.798	0.356	0.321
YOLO-v3 (coco)	3	0.072	0.148	0.090	0.174	0.338	0.240	0.009	0.011	0.0094	0.3502	0.047	0.084	3.270	0.381	0.313
Microsoft Computer Vision	5	0.079	0.129	0.126	0.193	0.218	0.214	0.018	0.011	0.0107	0.2657	0.069	0.110	3.229	0.370	0.324
ResNet50	5	0.015	0.123	0.023	0.181	0.054	0.208	0.005	0.013	0.0055	0.1163	0.016	0.028	3.629	0.411	0.369
ResNet50 (coco)	5	0.067	0.162	0.100	0.216	0.235	0.118	0.008	0.013	0.0089	0.3029	0.055	0.093	3.229	0.421	0.371
YOLO-v3	5	0.015	0.104	0.024	0.153	0.055	0.337	0.005	0.012	0.0053	0.1491	0.017	0.031	3.726	0.401	0.366
YOLO-v3 (coco)	5	0.071	0.159	0.100	0.206	0.273	0.259	0.009	0.013	0.0098	0.3343	0.056	0.096	3.231	0.411	0.356

Table 6: Results for the Microsoft Computer Vision, ResNet50 and YOLO-v3 APIs trained on the ImageNet and COCO dataset and evaluated on the Visual Genome dataset. The color gradient scales the APIs performance from green (best) to red (worst).

API	Accuracy	Accuracy (Semantic)	Recall	Recall (Semantic)	Precision	Precision (Semantic)	F1	F1 (Semantic)	Macro Precision	Macro Recall	Macro F1	Micro Precision	Micro Recall	Micro F1	WMD	Fine-Tuned BERT	Fine-Tuned RoBERTa
Clarifai	0.071	0.153	0.071	0.153	0.207	0.476	0.098	0.212	0.166	0.075	0.091	0.863	0.053	0.099	3.241	0.315	0.290
Google Cloud Vision	0.079	0.131	0.079	0.131	0.226	0.402	0.109	0.182	<b>0.278</b>	<b>0.118</b>	<b>0.145</b>	<b>0.896</b>	0.057	0.108	3.343	0.338	0.297
IBM Watson	0.038	0.109	0.038	0.109	0.121	0.401	0.054	0.158	0.106	<b>0.086</b>	<b>0.045</b>	0.820	0.031	0.059	3.379	0.206	0.198
Imgaga	0.071	0.142	0.071	0.142	0.218	0.431	0.099	0.198	0.207	0.104	0.120	0.787	0.055	0.103	3.420	0.322	0.290
Microsoft Computer Vision	<b>0.185</b>	<b>0.212</b>	<b>0.185</b>	<b>0.212</b>	<b>0.467</b>	<b>0.729</b>	<b>0.271</b>	<b>0.306</b>	0.242	0.116	0.142	0.807	<b>0.156</b>	<b>0.264</b>	<b>2.795</b>	<b>0.420</b>	<b>0.359</b>
Wolfram	0.046	0.107	0.046	0.107	0.187	0.391	0.069	0.158	0.094	0.072	0.070	0.786	0.047	0.090	3.629	0.282	0.244
DeepDetect	0.008	0.084	0.008	0.084	0.028	0.317	0.011	0.125	0.059	0.054	0.052	0.394	0.007	0.014	3.756	0.220	0.208
InceptionResNet-v2	0.016	0.105	0.016	0.105	0.054	0.370	0.023	0.152	0.094	0.108	0.092	0.505	0.014	0.027	3.673	0.250	0.230
Inception-v3	0.016	0.099	0.016	0.099	0.052	0.341	0.022	0.143	0.082	0.107	0.085	0.430	0.013	0.026	3.696	0.244	0.226
MobileNet-v2	0.012	0.095	0.012	0.095	0.044	0.334	0.018	0.138	0.083	0.074	0.074	0.411	0.011	0.022	3.744	0.232	0.217
Resnet50	0.014	0.090	0.014	0.090	0.044	0.318	0.019	0.130	0.071	0.091	0.074	0.431	0.011	0.022	3.760	0.226	0.216
VGG19	0.013	0.096	0.013	0.096	0.042	0.331	0.018	0.138	0.074	0.090	0.072	0.393	0.011	0.021	3.745	0.227	0.219
YOLO-v3	0.015	<b>0.079</b>	0.015	<b>0.079</b>	0.046	<b>0.275</b>	<b>0.020</b>	<b>0.114</b>	0.072	0.095	0.076	0.511	0.012	0.023	<b>3.806</b>	0.226	<b>0.209</b>

Table 7: The APIs’ metrics results with the top predicted label per image evaluated on the Open Images dataset. The color gradient scales the APIs performance from green (best) to red (worst).

API	Accuracy	Accuracy (Semantic)	Recall	Recall (Semantic)	Precision	Precision (Semantic)	F1	F1 (Semantic)	Macro Precision	Macro Recall	Macro F1	Micro Precision	Micro Recall	Micro F1	WMD	Fine-Tuned BERT	Fine-Tuned RoBERTa
Clarifai	0.098	0.202	0.160	0.313	0.172	0.350	0.152	0.300	0.253	0.157	0.172	0.783	0.131	0.225	3.014	0.383	0.342
Google Cloud Vision	0.118	0.194	0.190	0.292	0.188	0.316	0.172	0.276	<b>0.413</b>	<b>0.238</b>	<b>0.279</b>	<b>0.853</b>	0.142	0.244	3.044	0.417	0.367
IBM Watson	0.107	0.210	0.181	0.312	0.185	0.363	0.166	0.305	0.215	0.114	0.135	0.768	0.142	0.240	2.901	0.313	0.296
Imgaga	0.103	0.200	0.175	0.303	0.179	0.334	0.161	0.289	0.295	0.188	0.204	0.748	0.137	0.231	3.105	0.400	0.366
Microsoft Computer Vision	<b>0.188</b>	<b>0.288</b>	<b>0.240</b>	<b>0.346</b>	<b>0.452</b>	<b>0.622</b>	<b>0.284</b>	<b>0.405</b>	0.293	0.180	0.196	0.763	<b>0.204</b>	<b>0.322</b>	<b>2.645</b>	<b>0.470</b>	<b>0.407</b>
Wolfram	0.059	0.153	0.098	0.242	0.115	0.277	0.096	0.234	0.203	0.137	0.133	0.652	0.088	0.154	3.423	0.326	0.288
DeepDetect	0.009	0.136	0.015	0.200	0.019	0.258	0.015	0.206	0.047	0.093	0.054	0.272	0.014	0.027	3.594	0.258	0.249
InceptionResNet-v2	0.013	0.148	0.023	0.217	0.028	0.274	0.022	0.222	0.074	0.153	0.091	0.301	0.022	0.040	3.545	0.280	0.265
Inception-v3	0.013	0.145	0.023	0.218	0.028	0.272	0.022	0.220	0.074	0.152	0.089	0.286	0.021	0.039	3.545	0.274	0.258
MobileNet-v2	0.012	0.139	0.020	0.209	0.024	0.259	0.019	0.211	0.059	0.136	0.073	0.262	0.018	0.034	3.580	0.208	0.256
Resnet50	0.011	0.139	0.020	0.207	0.023	0.260	0.019	0.211	0.058	0.124	0.071	0.244	0.018	0.033	3.597	0.268	0.257
VGG19	0.011	0.137	0.020	0.206	0.022	0.253	0.018	0.207	0.057	0.121	0.069	0.228	0.017	0.031	3.592	0.264	0.254
YOLO-v3	0.011	<b>0.116</b>	0.019	<b>0.178</b>	0.022	<b>0.222</b>	0.018	<b>0.180</b>	0.052	0.117	0.065	0.264	0.017	0.032	<b>3.637</b>	0.277	0.261

Table 8: The APIs’ metrics results with the top three predicted labels per image evaluated on the Open Images dataset. The color gradient scales the APIs performance from green (best) to red (worst).

API	Accuracy	Accuracy (Semantic)	Recall	Recall (Semantic)	Precision	Precision (Semantic)	F1	F1 (Semantic)	Macro Precision	Macro Recall	Macro F1	Micro Precision	Micro Recall	Micro F1	WMD	Fine-Tuned BERT	Fine-Tuned RoBERTa
Clarifai	0.096	0.198	0.215	0.409	0.143	0.283	0.158	0.305	0.268	0.189	0.199	0.729	0.182	0.291	3.018	0.391	0.348
Google Cloud Vision	0.113	0.186	0.261	0.386	0.154	0.258	0.176	0.279	<b>0.501</b>	<b>0.327</b>	<b>0.367</b>	<b>0.816</b>	0.192	0.311	3.031	0.433	0.380
IBM Watson	0.116	0.213	0.264	0.421	0.170	0.307	0.188	0.322	0.239	0.159	0.175	0.718	0.212	0.328	2.859	0.339	0.317
Imgazg	0.110	0.197	0.257	0.412	0.160	0.275	0.180	0.301	0.318	0.245	0.251	0.715	0.203	0.316	3.033	0.424	0.385
Microsoft Computer Vision	<b>0.186</b>	<b>0.278</b>	<b>0.266</b>	<b>0.385</b>	<b>0.420</b>	<b>0.566</b>	<b>0.282</b>	<b>0.399</b>	0.315	0.213	0.224	0.714	<b>0.250</b>	<b>0.348</b>	<b>2.622</b>	<b>0.476</b>	<b>0.416</b>
Wolfarm	0.058	0.156	0.130	0.329	0.092	0.230	0.099	0.246	0.209	0.174	0.151	0.558	0.118	0.194	3.386	0.342	0.307
DeepDetect	<b>0.008</b>	<b>0.133</b>	<b>0.018</b>	<b>0.254</b>	<b>0.014</b>	<b>0.205</b>	<b>0.014</b>	<b>0.208</b>	0.041	0.107	0.052	0.208	0.017	0.032	3.580	0.263	0.252
InceptionResNet-v2	0.011	0.148	0.026	0.284	0.019	0.221	0.019	0.228	0.055	0.163	0.074	0.210	0.024	0.043	3.542	0.283	0.267
Inception-v3	0.011	0.145	0.025	0.282	0.019	0.219	0.019	0.226	0.056	0.165	0.074	0.206	0.024	0.042	3.540	0.278	0.261
MobileNet-v2	0.010	0.143	0.023	0.277	0.016	0.215	0.017	0.222	0.041	0.153	0.059	0.180	0.021	0.037	3.564	0.269	0.259
ResNet50	0.010	0.140	0.023	0.272	0.017	0.210	0.017	0.217	0.042	0.147	0.059	0.178	0.021	0.038	3.571	0.269	0.262
VGGL19	0.009	0.140	0.024	0.274	0.016	0.210	0.017	0.217	0.045	0.133	0.059	0.172	0.020	0.036	3.575	0.266	0.255
YOLO-v3	<b>0.009</b>	<b>0.120</b>	<b>0.021</b>	<b>0.239</b>	<b>0.015</b>	<b>0.181</b>	<b>0.016</b>	<b>0.189</b>	<b>0.038</b>	<b>0.128</b>	<b>0.053</b>	<b>0.188</b>	<b>0.019</b>	<b>0.035</b>	<b>3.628</b>	<b>0.284</b>	<b>0.266</b>

Table 9: The APIs’ metrics results with the top five predicted labels per image evaluated on the Open Images dataset. The color gradient scales the APIs performance from green (best) to red (worst).

API	Top Accuracy	Accuracy (Semantic)	Recall	Recall (Semantic)	Precision	Precision (Semantic)	F1	F1 (Semantic)	Macro Precision	Macro Recall	Macro F1	Micro Precision	Micro Recall	Micro F1	WMD	Fine-Tuned BERT	Fine-Tuned RoBERTa
Microsoft Computer Vision	1	0.185	0.212	0.185	0.212	0.667	0.271	0.306	0.242	0.116	0.147	0.8074	0.156	0.264	2.795	0.420	0.359
ResNet50	1	0.014	0.090	0.014	0.090	0.044	0.318	0.130	0.071	0.091	0.0736	0.4314	0.011	0.022	3.760	0.226	0.216
ResNet50 (coco)	1	0.161	0.197	0.161	0.197	0.021	0.241	0.230	0.052	0.075	0.0661	0.7090	0.148	0.244	2.910	0.381	0.330
YOLO-v3	1	0.015	0.079	0.015	0.079	0.046	0.275	0.020	0.114	0.072	0.095	0.5109	0.012	0.023	3.806	0.226	0.209
YOLO-v3 (coco)	1	0.178	0.212	0.178	0.212	0.685	0.266	0.310	0.060	0.075	0.0577	0.7605	0.159	0.262	2.806	0.401	0.339
Microsoft Computer Vision	3	0.188	0.288	0.240	0.346	0.452	0.284	0.405	0.293	0.180	0.1957	0.7626	0.204	0.322	2.645	0.470	0.407
ResNet50	3	0.011	0.139	0.020	0.207	0.023	0.260	0.019	0.058	0.121	0.0711	0.2489	0.018	0.033	3.597	0.268	0.257
ResNet50 (coco)	3	0.120	0.265	0.185	0.350	0.263	0.493	0.200	0.036	0.117	0.0495	0.5121	0.175	0.261	2.873	0.393	0.351
YOLO-v3	3	0.011	0.116	0.019	0.178	0.022	0.222	0.018	0.052	0.117	0.0652	0.2640	0.017	0.032	3.637	0.277	0.261
YOLO-v3 (coco)	3	0.138	0.274	0.196	0.342	0.329	0.554	0.223	0.054	0.124	0.0668	0.5853	0.180	0.275	2.797	0.419	0.368
Microsoft Computer Vision	5	0.186	0.278	0.266	0.385	0.420	0.566	0.282	0.315	0.213	0.2237	0.7135	0.220	0.348	2.622	0.476	0.416
ResNet50	5	0.010	0.140	0.023	0.272	0.017	0.210	0.017	0.042	0.147	0.0592	0.1781	0.021	0.038	3.571	0.269	0.262
ResNet50 (coco)	5	0.098	0.235	0.191	0.401	0.181	0.368	0.168	0.031	0.135	0.0447	0.4121	0.182	0.252	2.895	0.374	0.338
YOLO-v3	5	0.009	0.120	0.021	0.239	0.015	0.181	0.016	0.038	0.128	0.0528	0.1881	0.019	0.035	3.628	0.284	0.266
YOLO-v3 (coco)	5	0.122	0.252	0.200	0.369	0.268	0.459	0.201	0.052	0.139	0.0654	0.5216	0.185	0.274	2.798	0.406	0.367

Table 10: Results for the Microsoft Computer Vision, ResNet50 and YOLO-v3 APIs trained on the ImageNet and COCO dataset and evaluated on the Open Images dataset. The color gradient scales the APIs performance from green (best) to red (worst).

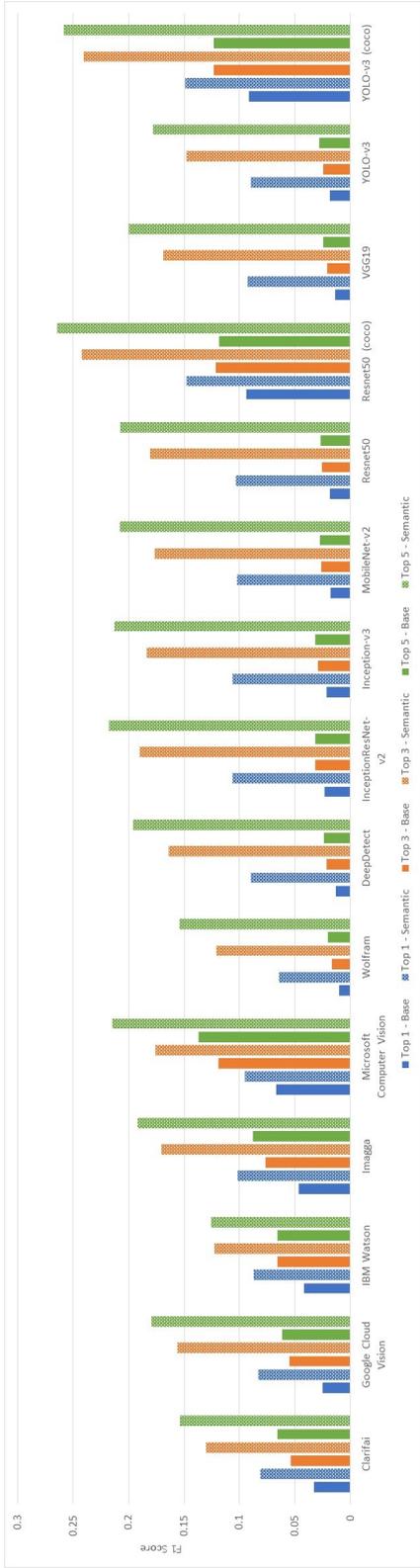


Figure 3: The  $F_1$  Results for each prediction level of the evaluated APIs on the Visual Genome dataset (higher scores are better).

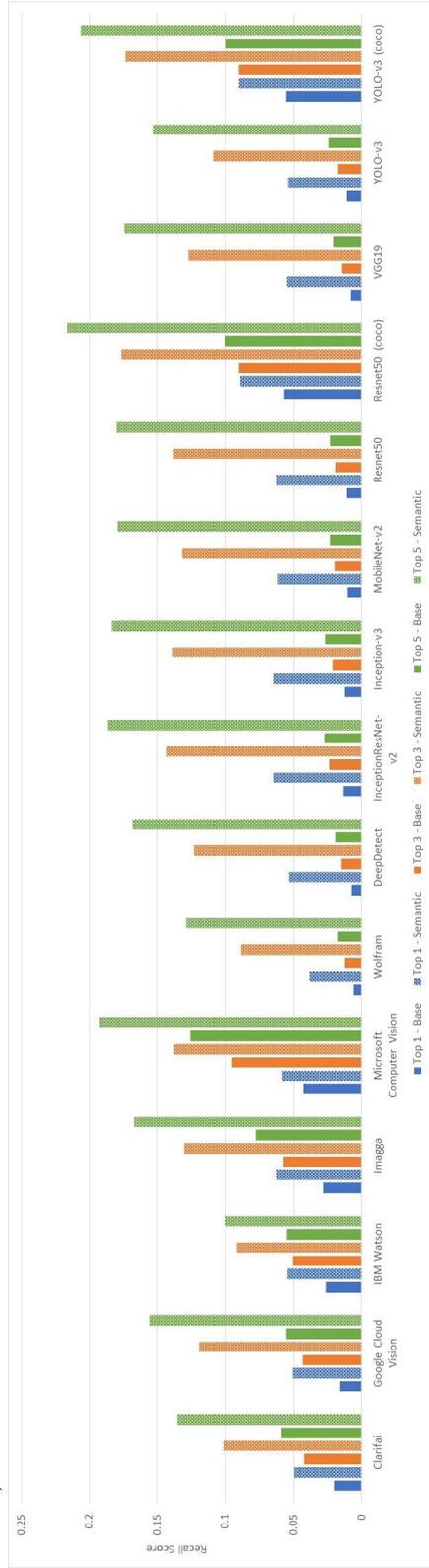


Figure 4: The *Recall* Results for each prediction level of the evaluated APIs on the Visual Genome dataset (higher scores are better).

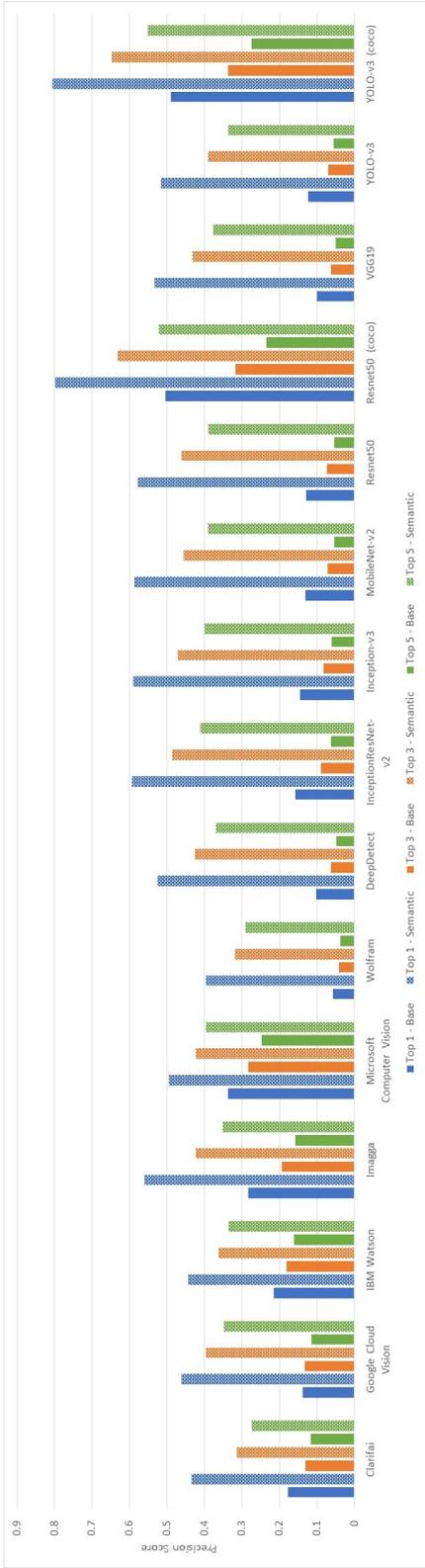


Figure 5: The *Precision* Results for each prediction level of the evaluated APIs on the Visual Genome dataset (higher scores are better).

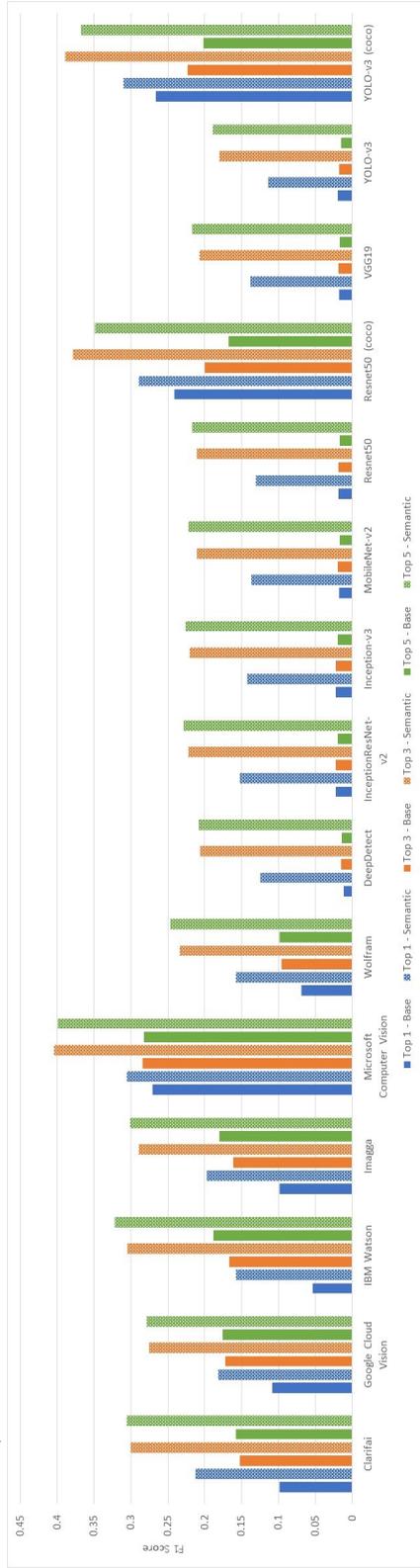


Figure 6: The  $F_1$  Results for each prediction level of the evaluated APIs on the Open Images dataset (higher scores are better).

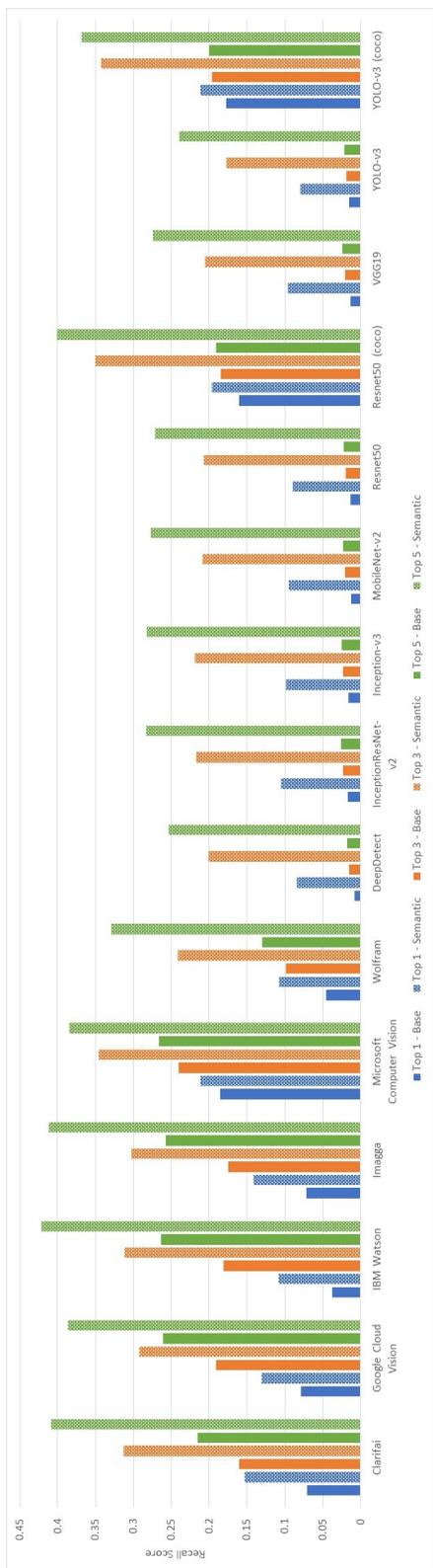


Figure 7: The *Recall* Results for each prediction level of the evaluated APIs on the Open Images dataset (higher scores are better).

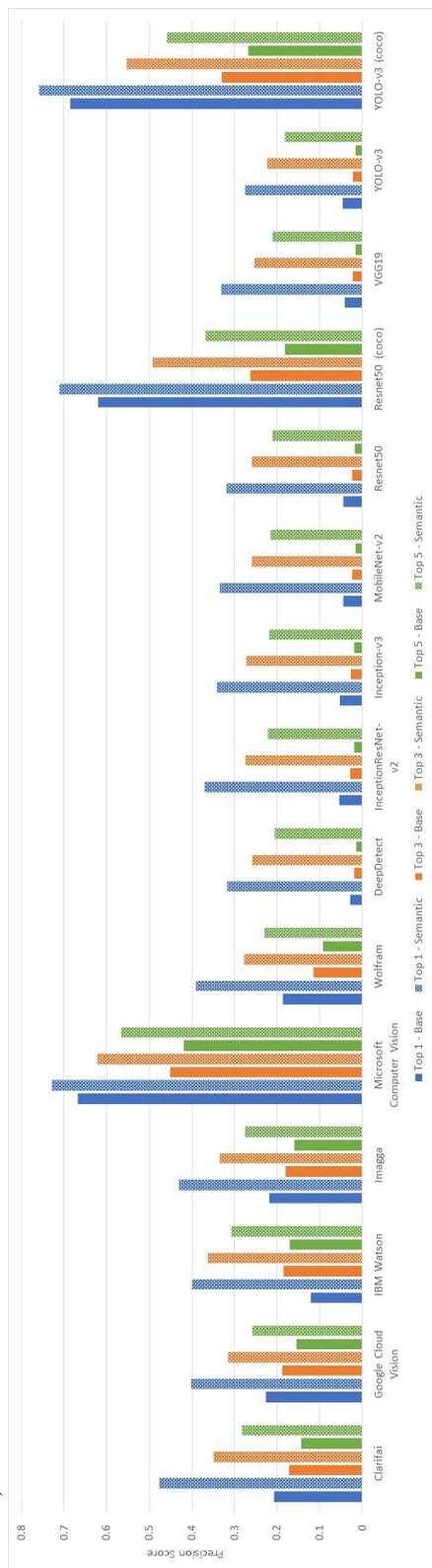


Figure 8: The *Precision* Results for each prediction level of the evaluated APIs on the Open Images dataset (higher scores are better).